CLAIMS

What is claimed is:

1. ~~A method for temporarily halting execution of a program's instructions while~~ the program is waiting for an event to occur, comprising:

arming an event monitor by identifying at least one event to be monitored;

requesting that the program be halted until any such identified event is observed by the event monitor; and

if execution of the program has been halted,

monitoring, by the event monitor, for an identified event; and

resuming execution of the program subsequent to observation of an identified event by the event monitor.

2. The method of Claim 1, further comprising:

halting execution of the program after requesting that the program be ~~halted if an identified event has not yet occurred since the arming.~~

3. The method of Claim 1 wherein identifying an event to be monitored comprises identifying at least one memory location to be monitored by the event monitor, and wherein the event comprises a modification to any such identified memory location.

4. The method of Claim 3 wherein the modification comprises a change of state.

5. The method of Claim 4 wherein a change of state comprises a change of access state.

6. The method of Claim 5 wherein a change of access state is from shared to exclusive.

Sub a7

~~7. The method of Claim 6 wherein a change of access state is observed by monitoring an inter-CPU messaging bus.~~

8. The method of Claim 4 wherein a change of state comprises a change of value.

9. The method of Claim 8, wherein a change in value is observed by monitoring a memory bus.

10. The method of Claim 8 wherein a change in value is observed as a write to the memory location.

Sub a7

~~11. The method of Claim 3, further comprising:~~

executing an arm instruction to arm the event monitor.

12. The method of Claim 11, wherein the arm instruction identifies the event to be monitored by identifying at least one memory location.

13. The method of Claim 12, wherein execution of the arm instruction further comprises:

recording a physical address of the memory location in a working register associated with the program, and

setting an indicator to a first state which enables the event monitor to monitor for the event, wherein the indicator is set to a second state if the event occurs.

~~while the program is halted,~~

fetching instructions from the program, and

allowing the fetched instructions to propagate into the instruction ~~pipeline.~~

23. The method of Claim 22, wherein instructions are fetched into an instruction buffer.

24. The method of Claim 1 wherein the program is executing in a multithreaded environment.

25. The method of Claim 24 wherein the environment is further a simultaneous multithreaded environment.

26. ~~The~~ method of Claim 1 wherein the program is executing in a multiprocessor environment.

27. ~~The method of Claim 1 further comprising the step of:~~

upon halting execution of the program, setting a timer to time a predetermined time interval, and starting the timer; and

resuming execution of the program upon expiration of a timer.

28. The method of Claim 27, further comprising:

stopping the timer if execution of the program is resumed due to observation of the event by the event monitor.

29. The method of Claim 1, wherein halting execution of the program results in a reduction of power consumption.

30. The method of Claim 1, wherein halting execution of the program allows other executing programs to utilize available resources.

31. A system for temporarily halting execution of a program's instructions while the program is waiting for an event to occur, comprising:

an event monitor which is armed via identification of an event to be monitored; and

an execution scheduler, responsive to the event monitor, which, upon a request that the program be halted until the event is observed by the event monitor, halts execution of the program if the event has not yet occurred since the event monitor was armed, and which resumes execution of the program upon observation of the event by the event monitor.

32. The system of Claim 31 wherein the event to be monitored is identified by at least one memory location to be monitored, and wherein the event comprises a modification to at least one of the identified memory locations.

33. The system of Claim 32 wherein the modification comprises a change of state.

34. The system of Claim 33 wherein a change of state comprises a change of access state.

35. The system of Claim 33 wherein the modification comprises a change of value.

36. The system of Claim 35 wherein a change in value is observed as a write to the memory location.

37. The system of Claim 32, further comprising:

an instruction flusher for flushing program instructions subsequent to the quiesce instruction from an instruction pipeline if the program is halted.

44. The system of Claim 43, wherein, while the program is halted, program instructions are fetched and allowed to propagate into the instruction pipeline.

45. The system of Claim 31 wherein the program is executing in a multithreaded environment.

46. The system of Claim 45 wherein the environment is further a simultaneous multithreaded environment.

47 The system of Claim 31 wherein the program is executing in a multiprocessor environment.

48. The system of Claim 31 further comprising:

a timer associated with the program such that a timer, upon the halting of its associated program, is set to time a predetermined time interval, and started, wherein execution of the program is resumed upon expiration of a timer; and wherein the timer is stopped if execution of the program is resumed due to observation of the event by the event monitor.

14. ~~The method of Claim 13 wherein the indicator is set to a second state if a change~~ to the memory location whose address is recorded in the working register is observed by the event monitor.

15. The method of Claim 13 wherein, in response to a request to quiesce, execution of the program is halted if the indicator is set to the first state.

16. The method of Claim 13 wherein, in response to a request to quiesce, execution of the program is not halted in response to the request to halt the program, if the indicator is set to the second state.

17. The method of Claim 13, wherein execution of the arm instruction further comprises:

loading a value from the identified memory location.

18. The method of Claim 13, further comprising:

executing a quiesce instruction to request that the program be halted.

19. The method of Claim 18 wherein in the arm instruction and queue instruction are assigned machine codes such that a program utilizing the instructions is functional if executed on a machine which does not support the instructions.

20. The method of Claim 18, further comprising:

flushing program instructions subsequent to the quiesce instruction from an instruction pipeline, upon halting execution of the program.

21. The method of Claim 3 wherein the modification is caused by another program.

22. ~~The method of Claim 1, further comprising:~~

~~a working register associated with the program, into which~~ a physical address of the memory location is stored upon arming of the event monitor; and

an indicator associated with the program, which is set to a first state upon arming the event server, causing the event monitor to monitor for the event, and which set to a second state by the event monitor upon a change to the memory location whose address is recorded in the working register.

38. The system of Claim 37, wherein a lock value is loaded from the identified memory location upon storing the memory location's address in the working register, such that a determination may be made as to whether the memory location's state has changed after arming the event monitor and before halting the program's execution.

39. The system of Claim 38 wherein an executing arm instruction arms the event monitor.

40. The system of Claim 39, wherein an executing quiesce instruction halts execution of the program only if the flag is set.

41. The system of Claim 40 wherein a change to the memory location comprises a change of state of the memory location, caused by another program, from shared to exclusive.

42. The system of Claim 40, wherein a change to the memory location comprises a write operation to the memory location, observed by monitoring the address on a memory write bus.

43. The system of Claim 40, further comprising: